

Sprawozdanie z pracy projektowej: Specyfikacja Systemu Magazynowego

Sara Fijołek
Dawid Bazylewicz
Adrian Sitnicki
Gerard Hagel

11 czerwca 2025

Spis treści

1	Opis funkcjonalny systemu	3
2	Opis technologiczny	5
3	Podział obowiązków i odpowiedzialności w zespole	5
4	Instrukcje lokalnego i zdalnego uruchomienia systemu	6
5	Wnioski projektowe	6
6	Repozytoria projektowe	7

1 Opis funkcjonalny systemu

System magazynowy służy do automatyzacji zarządzania zasobami magazynowymi, obejmując procesy logowania, zarządzania użytkownikami, monitorowania stanów magazynowych oraz obsługi operacji zarówno w trybie online, jak i offline.

L.p.	Funkcjonalność	API	Web	Mobile	Desktop
1	Automatyczne dodanie pierwszego administratora	X			
2	Logowanie administratora	X	X		X
3	Tworzenie nowych użytkowników przez administratora	X	X		X
4	Logowanie użytkownika	X		X	
5	Resetowanie hasła		X	X	X
6	Wylogowywanie się		X	X	X
7	Wyświetlanie stanu magazynowego	X	X		X
8	Modyfikacja stanu magazynowego	X	X		X
9	Generowanie raportów magazynowych	X	X		X
10	Powiadomienia przypominające o raporcie	X	X		X
11	Skanowanie kodu kreskowego, podanie ilości i lokalizacji	X		X	
12	Podgląd listy produktów w magazynie	X		X	
13	Wyszukiwanie lokalizacji produktu	X		X	
14	Kolejkowanie zadań w trybie offline	X		X	X
15	Pobranie bazy danych do użytku offline	X		X	X
16	Dodawanie nowych przedmiotów	X	X		X
17	Obsługa języka polskiego i angielskiego		X	X	X
18	Powiadomienie o przywróceniu połączenia i synchronizacji	X		X	

2 Opis technologiczny

Projekt wykorzystuje następujące technologie:

- **Aplikacja desktop:** C#, .NET
- **Aplikacja webowa:** TypeScript, React
- **API:** PHP, Laravel, MySQL
- **Aplikacja mobilna:** Dart, Flutter

3 Podział obowiązków i odpowiedzialności w zespole

- **Sara Fijołek** – frontend aplikacji webowej, implementacja interfejsów w React
- **Dawid Bazylewicz** – projekt i implementacja API, konfiguracja bazy danych MySQL
- **Adrian Sitnicki** – rozwój aplikacji desktopowej w C# i .NET
- **Gerard Hagel** – rozwój aplikacji mobilnej w Dart/Flutter

4 Instrukcje lokalnego i zdalnego uruchomienia systemu

- API:

Lokalne uruchomienie API:

Pobierz projekt z repozytorium projektowego.

Aplikacja jest skonteneryzowana, do uruchomienia otwórz terminal Dockera, przejdź do ścieżki pobranego projektu i wpisz:

```
docker compose up -d
```

Przy pierwszym uruchomieniu zainstaluj wszystkie zależności:

```
cd laravel; composer install
```

Jeśli route'y nie ustawiły się automatycznie, wejdź w kontener warehouse-laravel > Exec i wpisz:

```
a2enmod rewrite; service apache2 restart
```

- Aplikacja webowa:
Aby uruchomić aplikację webową lokanie na swoim komputerze należy pobrać api z repozytorium oraz pobrać frontend z repozytorium.
- Aplikacja desktopowa:
Brak
- Aplikacja mobilna::
Brak

5 Wnioski projektowe

- Dawid Bazylewicz (API):
Większość założeń projektowych zostało spełnione. Niestety, zabrakło nam czasu na dokończenie projektu. Brakujące funkcjonalności oraz wdrożenie łatwo można by było rozwinąć w tydzień. Projekt łatwo można rozbudować o dodatkowe funkcje, które nie zostały przedstawione w funkcjonalnościach. Pierwszy raz pisałem testy i myślę, że wyszło nie najgorzej.

- Sara Fijolek (Web):
Nie wszystko mi się udało zrobić, bo jeden problem mnie przerósł. Nauczyłam się jak ważna jest współpraca w grupie i jak sugestie kolegi mogą pomóc w rozwiązaniu problemu w kodzie. Oraz że wzorce projektowe pomagają w utrzymaniu czystości kodu.
- Adrian Sitnicki (Desktop):
Brak
- Gerard Hagel (Mobile):
Polecam nie zabierać się do pracy dwa dni przed oddaniem projektu.

6 Repozytoria projektowe

- Aplikacja webowa (Sara Fijolek):
https://github.com/SaraFijolek/Zmp_project.git
- Aplikacja desktopowa (Adrian Sitnicki):
<https://github.com/ASitnicki/WarehouseDesktop>
- Aplikacja mobilna (Gerard Hagel):
<https://github.com/GerardHagel/MagazineManager>
- Backend (Dawid Bazylewicz):
<https://github.com/DawBaz15/Projekt-Zmp>